Development of a Prototype Traffic Congestion Charging System in the Philippine Setting

Sean Francis Natividad, Ed Lorence de Guzman, Mikhail Ulanday, and Melvin Cabatuan*
Department of Electronics and Communications Engineering, De La Salle University
Manila, PHILIPPINES 1004
seanfrancisn@gmail.com, edlorencedg@gmail.com, gorbyto0904@gmail.com,
*melvin.cabatuan@dlsu.edu.ph

ABSTRACT

Congestion charging or the area based road-user charging policy around congested city sections has been shown to mitigate traffic congestion in modern cities, i.e. Singapore and London. In this paper, we present the design and development of an automated traffic congestion charging system. In particular, we demonstrate the implementation of automatic license plate recognition (ALPR) for Philippine plates using OpenALPR with the Tesseract OCR (optical character recognition) library as the congestion charging front-end. Then, a congestion charge collection system equipped with car database, registration, and auto-debit mechanism was also developed. The results illustrated an average plate recognition accuracy 87 % as tested on 0-20 kph and 20-40 kph speed ranges.

Keywords-*Congestion charging system; automatic license plate recognition(ALPR); OpenALPR; automatic number plate recognition (ANPR); optical character recognition (OCR) for Philippine cars*

I. INTRODUCTION

Congestion charging has been shown to improve problematic traffic conditions [1-2]. After its first of implementation in London, congestion decreased by 30 percent while overall traffic levels within the charging zone fell by 16 percent [1]. A similar effect was observed in Stockholm where improvements in travel times were large enough to be noticeable to the general public during congestion charging trials [2]. An enabling technology for congestion charging is automatic license plate recognition (ALPR). ALPR is the computer-aided acquisition of the vehicle license plate information from an image, image sequence, or video capture. The ALPR process principally includes image or video capture, feature extraction, plate detection & segmentation, and character extraction & recognition [3]. A comprehensive survey of state-of-the-art ALPR techniques is given in [3]. In this paper, we present the design and development of an automated traffic congestion charging system. In particular, we implemented and characterized open ALPR [4] for Philippine plates which serves as the front-end mechanism of the whole system. Furthermore, an integrated congestion charge collection system equipped with car database, registration, and auto-debit mechanism was also developed.

The rest of the paper is organized as follows – the methodology is discussed in the next section, while the data and results are elaborated in section III; finally, the paper is concluded in section IV.

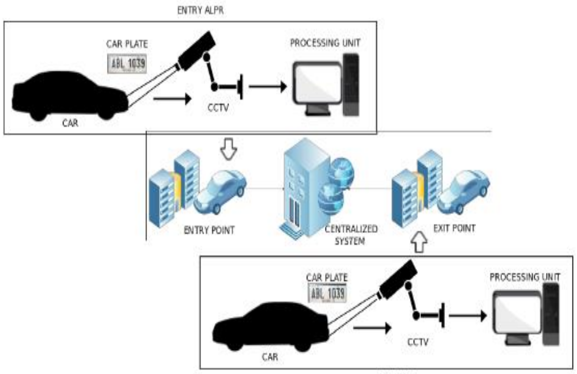

Fig. 1 Traffic Congestion Charging System Conceptual Diagram

II. METHODOLOGY

The overall conceptual diagram of the traffic congestion charging system is shown on Fig. 1. The congestion charging entry point system extracts license plate number from the passing vehicles. Then, the sends the extracted plates to a central office that processes information for debiting the associated vehicle accounts. Both entrance and exit points will send the gathered information of the passing vehicle and the central office will be given the task to determine which account the vehicle is linked to and debit the account for the corresponding congestion charge. Though the information processing is conducted at the central office, the plate number recognition will occur at the entry or exit point so that information transfer will be minimal. Thus, both the entry and exit point sub-systems would require a CCTV camera for video capture of the passing vehicles and a processing unit for plate number recognition. The entry and exit point subsystems are connected to the centralized system through a wired communication link or through the internet cloud server. It should be noted that the congestion charges associated to vehicle accounts are to be paid in a prepaid or post-payment arrangements, though a full blown execution of these payment schemes is beyond the scope of this study.

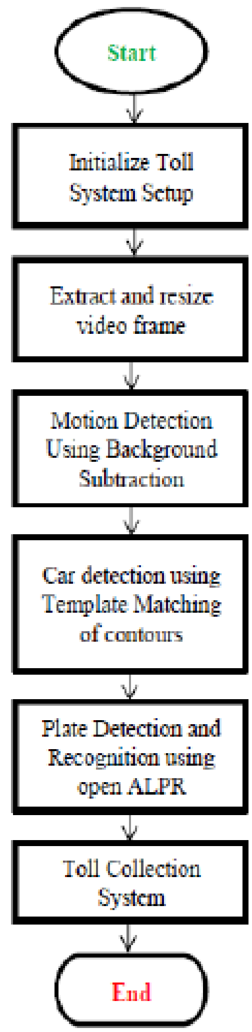

Fig. 2 Traffic Congestion Charging System Process Diagram

The traffic congestion charging system must first be initialized by selecting what video source to use, whether a video file for testing or through an IP camera for real-time video capture. Consequently, the system mode of operation is selected to be either entry or exit mode. Then, the traffic congestion charging location is also set. Afterwards, the system will extract and resize the frames for further image processing. After acquiring the necessary input, the system detects the presence of a moving object and determines whether the object is car. In the frame where the car is detected, the plate number will then be localized for plate number recognition. Finally, the extracted plate number is considered as input to the congestion charging system for the auto-debit of registered accounts.

Fig. 3 shows the process of detecting a car that will trigger the ALPR module. Resizing of the image frames is necessary for the purpose of making the process faster due to limitations in the processing unit. During calibration, the median of all the resized image frames comprises the background image. Once the background image has been computed, subtracting the current image to the background image would produce the blob or contour of the image where black pixels represent the areas without motion, while the white pixels generally represent movement, respectively. Afterwards, contour thresholding filters out small variations that usually signify minor motions or noisy illumination variations. Background subtraction is an approach widely used for identifying moving objects in videos from stationary cameras. One of the methods of background subtraction is frame difference method [5,6]. The concept of this approach involves the use of a reference frame, commonly called as the background frame, or background model. The current frame is compared to the reference frame and the detection of the moving image is from the difference between these two frames. The moving object or the foreground is set to be the white pixels while the nonmoving objects or the background is composed of black pixel.

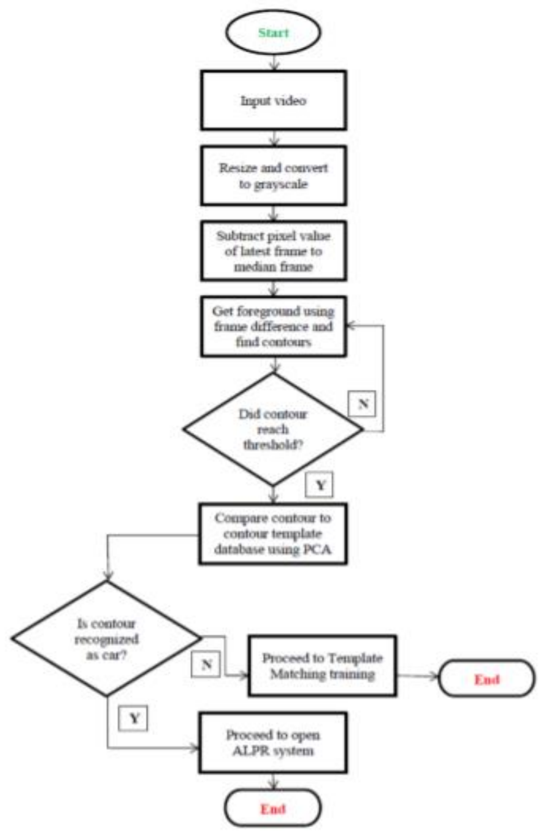

Fig. 3 Traffic Congestion Charging System Simple Car Detection Process

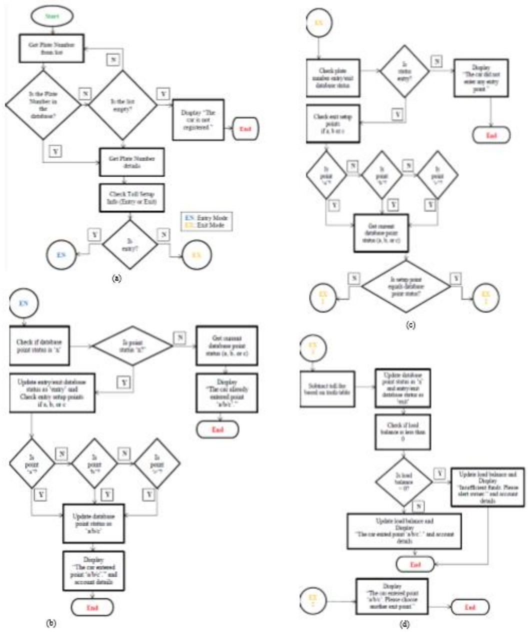

Fig. 4 Traffic Congestion Charging System Flowchart

Template matching involves searching for the position of a sub-image (i.e. template) within a whole image. Features that are extracted from object boundaries are compared to initially stored and identified features for detection. This approach can also be called features matching [7]. In the system, the contour of the foreground (car silhouette) is the feature that is matched to the features of the template database. In this study, we utilized an open source license plate recognition system called OpenALPR for the necessary plate number detection and recognition [4]. OpenALPR has its own plate number classifier for its plate detection process. This classifier model was used in order to find the plate number in the image by comparing it to the image. For its plate recognition process, it uses an additional character recognition library. OpenALPR uses LBP cascade to localize the location of the plate. After the plate recognition part, the automated congestion charging system executes the necessary fee deduction on the corresponding account. Given a recognized plate number, the system will deduct the congestion fee for a certain entry and exit point where the vehicle was detected. Figure 4 shows the flow chart of the congestion charging system and the entry is represented with EN while exit with EX. First, each of the recognized plate number from the open ALPR system plate recognition list is checked whether it exists in the database. Once a match has been found, the plate number details in the account database will be prepared. Then the system will check whether the congestion charging set-up is in entry mode or exit mode. The default entry/exit status in the database is 'exit' and the default point status in the database is 'x'. If the system is in entry mode, it will check whether the point status of the plate number in the database is 'x'. If it is 'x', the system will update the entry/exit status in the database as 'entry', check whether the entry point of the toll set up is 'a', 'b' or 'c' and update the point status in the database based on the set-up and display the necessary details. However, if the point status in the database is not 'x', this means that the car already entered the system. The system will get the current point status in the database and display what point the car entered. On the other hand, if the system is in exit mode, it will check the entry/exit status of the car in the database. If the status is 'exit', the system will display that the car did not enter any exit point. If it is 'entry', the system will check the exit point of the toll set up, compare it to the point status of the plate number in the database and display that the car already entered the same point if they are equal.

However, if they are not equal, the system will subtract the toll fee. After knowing the desired toll fee, the system will update the point status and entry/exit status back to default, check the load balance whether it is less than 0 or not and display the necessary details which ends the congestion charge collection system. The congestion charge collection will depend on the combination of the entry and exit points. The status column is the assumed status of the vehicle. If the vehicle enters point A and exits at point B, it will result to a status of AB with a toll fee set to 7 while AC if it exits at point C with at oll fee set to 11. If the vehicle enters point B and exits at point A, it will result to a status of AB with a toll fee set to 7 while BC if it exits at point C with a toll fee set to 9. If the vehicle enters point C and exits at point A, it will result to a status of AC with a toll fee set to 11 while BC if it exits at point B with a toll fee set to 9. Since a vehicle cannot enter and exit at a given point at the same time, it will result to an error status.

Fig. 5 Congestion charging ALPR experimental set-up: (a) Data Gathering ; (b) LBP cascade classifier license plate detection sample

The system set-up for data gathering of video feeds is shown in the Fig. 5. The IP camera captures the video, and sends it to the PC via the router. The IP camera is connected through a LAN cable while the laptop is connected to the WiFi router to connect to the IP address of the IP Camera. The IP camera is placed on the stand which is approximately 3 ft. high is directly placed in front of the car's path approximately 33.5o, wherein the distance from the camera to the estimated optimal reading position is around 5.25 meters. In order to validate the speed of the cars as data for the system, the researchers recorded the video of the speedometer of the car for each plate of each speed range simultaneously while the car is being recorded by the IP Camera. Fig. 5-a shows a small screenshot of the video file of the speedometer of the car.

III.DATA AND RESULTS

A. *Initial Experimentation*

Thirty (30) Philippine plate number replicas were printed on a cardboard for the purpose of experimentation. These fabricated plate number replicas will be attached to the plate number section of the test vehicle. Fig. x illustrates the resemblance between the plate replica on top (AKA5654) and the actual plate number (ABL1039) below it.

Fig. 6 Example Philippine plate number replica on top (AKA5654) with the actual license plate (ABL1039) on the bottom

Template matching was utilized for car detection. Around 177 foreground images where initially extracted through background subtraction which were consequently considered as templates for the system to detect a moving vehicle in a video. This ensures detection of a car-like silhouette as opposed to other distinct moving objects that might falsely trigger the system. The templates were bitmap images consisting of black and white pixels based from the foreground vehicle. The system will examine each foreground from the video stream and compare it to the template to determine whether the moving object is a car through correlation. Sample car templates are shown in Fig. x. Whenever the contour of a specific frame reaches a certain threshold, the system will compare it to the template database with the aid of principal component analysis (PCA). PCA transforms possible correlated variables into principal components and reduces high-dimensional data space (set of templates) into a lower-dimensional picture and compares the image with the templates by calculating the Euclidean distances between the image and the templates [7]. If the contour is not recognized as a car, the image is saved in to be manually added to the templates while if recognized, the system will just continue to recognize foreground contours. In the proposed system, the feature utilized for car detection was the size of the contour of the moving object. Whenever the contour of the moving object reaches a certain threshold value, the system will compare the frame to the said templates using the *EigenObjectRecognizer* function of

Fig. 7 Example vehicle silhouette as detection templates

EmguCV which correlates the images to distinguish whether the contour is that of a car. If the given image is confirmed to be a car, the system will proceed to the openALPR system. If it does not recognize the contour to be a car, the system saves the image of the contour so that it can be added to the templates if the user wants it to be recognized as a car, else, it will be discarded. Plate detection utilized the openALPR library which incorporates the necessary trained model for the plate detection of the system. The system uses a trained xml model to detect the plate number inside a frame. It will also take the plate number region-of-interest (ROI) for plate recognition. For the plate detection results of 0-20kph speed range, all detections were successful. This means that out of all the 30 plate numbers tested, all of them were detected by the openALPR system. Meanwhile, the 20-40kph and 40-60kph speed ranges each had one undetected plate number, which means that out of 30 plates, it only detected 29 plates. openALPR also has its built-in plate recognition capability with Tesseract OCR libraries. It has its own file type ".traineddata" which is the trained model for character recognition of the system. This will then be utilized to locate and recognize the characters inside a region of interest which in this case represents the detected plate number. The openALPR system outputs a list of all possible plate number recognition. There are instances wherein the correct recognition result did not have the highest confidence level. In order to select the plate number, the system matches each recognized plate number to the given plate numbers in the database and selects the plate when a matched is given. For the 0-20kph speed range, 27 out of 30 plates were recognized correctly, for the 20-40kph speed range, 24 out of 30 plates were recognized correctly and finally, for the 40-60kph speed range, 26 out of 30 plates were recognized correctly.

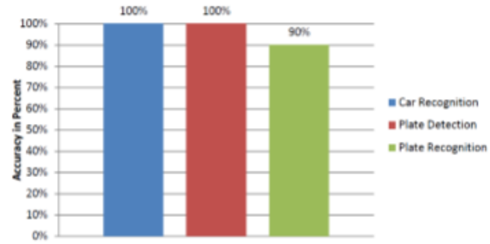

Fig. 8 Accuracy for 0-20KPH Speed Range Data Set

Fig. 8 – 10 shows the accuracy of each speed range based from the car detection system and plate detection and plate recognition of the OpenALPR system. Based on Figure x, the car detection, plate detection, and plate recognition systems each has an accuracy of 100%, 100% and 90% respectively for the 0-20kph speed range data set. Based on Figure xx, the car detection, plate detection, and plate recognition systems each has an accuracy of 100%, 96.67% and 80% respectively for the 20-40kph speed range data set. Based on Figure xxx, the car detection, plate detection, and plate recognition systems each has an accuracy of 100%, 96.67% and 86.67% respectively for the 0-20kph speed range data set. It is noticeable that the 40 to 60 KPH accuracy is higher as compared with 20 to 40 KPH. This might be attributed to the illumination of the area during experimentation. The 20 to 40 KPH data was gathered around 3:00 PM to 6:00 PM where the illumination is nearing its lowest point, while the 40 to 60 KPH was gathered during 12:00 PM to 3:00 PM where the illumination is at its peak giving the system a higher chance to recognize the plate.

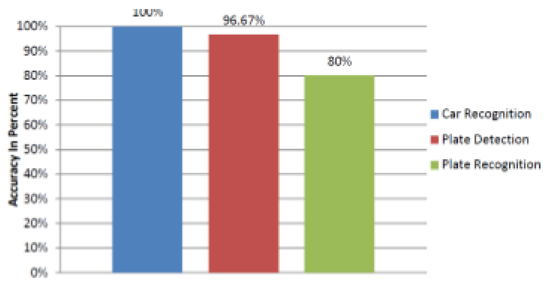

Fig. 9 Accuracy for 20-40KPH Speed Range Data Set

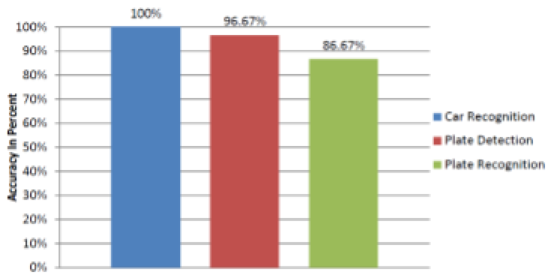

Fig. 10 Accuracy for 40-60KPH Speed Range Data Set

B. Repeatability Tests

Repeatability tests were also conducted wherein test runs of the same plate for 5 times for each car entry into the system zone and 5 times for each exit from the system zone to determine whether the system will still have a stable accuracy. The plate recognition accuracy for the repeatability test was able to reach a 91.33% accuracy which is tabulated in Table x. The false positives were mostly due to misreading of one or more character/s in the plate. These might be caused by the changes of angle during the passing of the vehicle, illumination of the day, and processing capacity of the machine

TABLE I
PLATE DETECTION REPEATABILITY TEST ACCURACY FOR 0-20 KPH FOR ENTRANCE

Plate Detection (Entrance)					
	Plate No.	True Positive	False Positive	False Negative	Result Percentage
1	AAA6088	5	0	0	100
2	AAB9669	5	0	0	100
3	AAC7171	5	0	0	100
4	AAD1159	5	0	0	100
5	AAJ8762	4	1	0	80
6	AAM1231	5	0	0	100
7	AAN1955	5	0	0	100
8	AAP7860	5	0	0	100
9	AAQ6593	5	0	0	100
10	AAW3081	5	0	0	100
11	AAY3227	5	0	0	100
12	AAZ3545	5	0	0	100
13	ABF7348	5	0	0	100
14	ABG6025	5	0	0	100
15	ABL2841	5	0	0	100
16	ABL8929	5	0	0	100
17	ACA6464	5	0	0	100
18	ADA1716	5	0	0	100
19	AEA9839	5	0	0	100
20	AHA8928	5	0	0	100
21	AIA8930	5	0	0	100
22	AJA3359	5	0	0	100
23	AKA5654	5	0	0	100
24	AOA9539	5	0	0	100
25	ARA1577	5	0	0	100
26	ASA1716	4	1	0	80
27	ATA6668	5	0	0	100
28	AUA9975	5	0	0	100
29	AVA2579	5	0	0	100
30	AXA1834	5	0	0	100
	Total	148	2		98.66667

Tesseract is dependent on the processing capacity of the machine to give out a more accurate reading. If the machine would have a higher processing capacity, the Tesseract would no longer need to reduce the image size for faster output. Most of the misreading of the system involves the following, namely, 6 instead of 8, I instead of D, and 1 instead of I. Another factor that contributed for the false positive of recognition was because of the plate detection had also reduced to 98.67% accuracy causing 2 of the misreading or no reading from the plate recognition system which could be seen from Table I.

TABLE II
PLATE RECOGNITION REPEATABILITY TEST ACCURACY FOR 0-20 KPH FOR ENTRANCE

Plate Recognition (Entrance)					
	Plate No.	True Positive	False Positive	False Negative	Result Percentage
1	AAA6088	3	2	0	60
2	AAB9669	5	0	0	100
3	AAC7171	4	1	0	80
4	AAD1159	5	0	0	100
5	AAJ8762	4	1	0	80
6	AAM1231	5	0	0	100
7	AAN1955	5	0	0	100
8	AAP7860	5	0	0	100
9	AAQ6593	5	0	0	100
10	AAW3081	3	2	0	60
11	AAY3227	5	0	0	100
12	AAZ3545	5	0	0	100
13	ABF7348	5	0	0	100
14	ABG6025	5	0	0	100
15	ABL2841	5	0	0	100
16	ABL8929	5	0	0	100
17	ACA6464	4	1	0	80
18	ADA1716	5	0	0	100
19	AEA9839	5	0	0	100
20	AHA8928	5	0	0	100
21	AIA8930	4	1	0	80
22	AJA3359	5	0	0	100
23	AKA5654	5	0	0	100
24	AOA9539	2	3	0	40
25	ARA1577	5	0	0	100
26	ASA1716	4	1	0	80
27	ATA6668	4	1	0	80
28	AUA9975	5	0	0	100
29	AVA2579	5	0	0	100
30	AXA1834	5	0	0	100
	Total	137	13		91.33333

While on the exit phase setup the system was able to score a 90% accuracy which is lower by 1.33% or equivalent to 2 trials of misread or unrecognized plates which can be seen if Table II and Table IV is compared. The accuracy of the plate detection also decreased which can be seen if Table I and Table III is compared whereas 1 plate has not been detected from the 150 data that was gathered from the exit phase of 0 to 20 KPH testing. The factor that caused the decrease of accuracy that the researcher are gradually considering is the change of illumination during data gathering. Wherein the 0 to 20 KPH entrance data was gathered during the afternoon where illumination was considered to be at its peak while the exit data was gathered during early morning where illumination was considered to be decreased as compared to the entrance data. Illumination could be considered to be a gradual factor for the system as the plate characters is highly contrasting to the background of the plate where changes of illumination could affect the output of the system. The lowest plate recognition accuracy for the repeatability test for the 20 to 40 KPH phase is 80% which is shown in Table VII. Another aspect from the 20 to 40 KPH that may be noticeable is the high difference of accuracy percentage of recognition from the entrance phase to exit phase. The probable cause of the high decrease of accuracy was a huge change of illumination due to the location change of the

TABLE III
PLATE DETECTION REPEATABILITY TEST ACCURACY FOR 0-20 KPH FOR EXIT

Plate Detection (Exit)					
	Plate No.	True Positive	False Positive	False Negative	Result Percentage
1	AAA6088	5	0	0	100
2	AAB9669	5	0	0	100
3	AAC7171	5	0	0	100
4	AAD1159	5	0	0	100
5	AAJ8762	5	0	0	100
6	AAM1231	5	0	0	100
7	AAN1955	5	0	0	100
8	AAP7860	5	0	0	100
9	AAQ6593	5	0	0	100
10	AAW3081	5	0	0	100
11	AAY3227	5	0	0	100
12	AAZ3545	5	0	0	100
13	ABF7348	5	0	0	100
14	ABG6025	5	0	0	100
15	ABL2841	5	0	0	100
16	ABL8929	5	0	0	100
17	ACA6464	4	1	0	80
18	ADA1716	5	0	0	100
19	AEA9839	4	1	0	80
20	AHA8928	5	0	0	100
21	AIA8930	5	0	0	100
22	AJA3359	5	0	0	100
23	AKA5654	5	0	0	100
24	AOA9539	5	0	0	100
25	ARA1577	5	0	0	100
26	ASA1716	5	0	0	100
27	ATA6668	5	0	0	100
28	AUA9975	4	1	0	80
29	AVA2579	5	0	0	100
30	AXA1834	5	0	0	100
	Total	147	3	0	98

TABLE IV
PLATE DETECTION REPEATABILITY TEST ACCURACY FOR 0-20 KPH FOR ENTRANCE

Plate Recognition (Exit)					
	Plate No.	True Positive	False Positive	False Negative	Result Percentage
1	AAA6088	3	2	0	60
2	AAB9669	3	2	0	60
3	AAC7171	5	0	0	100
4	AAD1159	4	1	0	80
5	AAJ8762	5	0	0	100
6	AAM1231	5	0	0	100
7	AAN1955	5	0	0	100
8	AAP7860	5	0	0	100
9	AAQ6593	5	0	0	100
10	AAW3081	4	1	0	80
11	AAY3227	5	0	0	100
12	AAZ3545	5	0	0	100
13	ABF7348	5	0	0	100
14	ABG6025	5	0	0	100
15	ABL2841	5	0	0	100
16	ABL8929	5	0	0	100
17	ACA6464	3	2	0	60
18	ADA1716	5	0	0	100
19	AEA9839	4	1	0	80
20	AHA8928	5	0	0	100
21	AIA8930	1	4	0	20
22	AJA3359	5	0	0	100
23	AKA5654	5	0	0	100
24	AOA9539	4	1	0	80
25	ARA1577	5	0	0	100
26	ASA1716	5	0	0	100
27	ATA6668	5	0	0	100
28	AUA9975	4	1	0	80
29	AVA2579	5	0	0	100
30	AXA1834	5	0	0	100
	Total	135	15	0	90

entrance and exit setup. It also noticeable that the illumination issue did not affect the plate detection system too much as compared with the recognition. This only gives more weight to consider that illumination is one of the notable factor that could affect the plate recognition system. This is because of the plate character and background contrast which gradually affect the recognition output and accuracy.

TABLE V
PLATE DETECTIONS REPEATABILITY TEST ACCURACY FOR 20-40 KPH FOR ENTRANCE

Plate Detection (Entrance)					
	Plate No.	True Positive	False Positive	False Negative	Result Percentage
1	AAA6088	5	0	0	100
2	AAB9669	5	0	0	100
3	AAC7171	5	0	0	100
4	AAD1159	5	0	0	100
5	AAJ8762	5	0	0	100
6	AAM1231	5	0	0	100
7	AAN1955	5	0	0	100
8	AAP7860	5	0	0	100
9	AAQ6593	3	2	0	60
10	AAW3081	5	0	0	100
11	AAY3227	5	0	0	100
12	AAZ3545	5	0	0	100
13	ABF7348	5	0	0	100
14	ABG6025	5	0	0	100
15	ABL2841	5	0	0	100
16	ABL8929	5	0	0	100
17	ACA6464	5	0	0	100
18	ADA1716	5	0	0	100
19	AEA9839	4	1	0	80
20	AHA8928	5	0	0	100
21	AIA8930	5	0	0	100
22	AJA3359	4	1	0	80
23	AKA5654	4	1	0	80
24	AOA9539	5	0	0	100
25	ARA1577	4	1	0	80
26	ASA1716	5	0	0	100
27	ATA6668	5	0	0	100
28	AUA9975	5	0	0	100
29	AVA2579	5	0	0	100
30	AXA1834	5	0	0	100
	Total	144	6	0	96

C. Graphic User Interface

Figure 11 shows the main GUI of the system. The video stream image box in the leftmost part shows the input video stream of the system. When a moving object is recognized to be a car, a bounding box will appear. The license plate ROI image box shows the detected plate number from a recognized car. The plate list box under the license plate ROI image box shows the possible recognition results of the openALPR. The Note group box shows the status of the car detection and plate recognition. The Car Information group box has the following text boxes and buttons and their functions:

- Plate Number – shows the recognized plate number that matched from the car information database.

TABLE VI
PLATE RECOGNITION REPEATABILITY TEST ACCURACY FOR 20-40 KPH FOR ENTRANCE

Plate Recognition (Entrance)					
	Plate No.	True Positive	False Positive	False Negative	Result Percentage
1	AAA6088	5	0	0	100
2	AAB9669	5	0	0	100
3	AAC7171	4	1	0	80
4	AAD1159	4	1	0	80
5	AAJ8762	5	0	0	100
6	AAM1231	4	1	0	80
7	AAN1955	5	0	0	100
8	AAP7860	5	0	0	100
9	AAQ6593	3	2	0	60
10	AAW3081	5	0	0	100
11	AAY3227	5	0	0	100
12	AAZ3545	5	0	0	100
13	ABF7348	5	0	0	100
14	ABG6025	4	1	0	80
15	ABL2841	4	1	0	80
16	ABL8929	5	0	0	100
17	ACA6464	3	2	0	60
18	ADA1716	5	0	0	100
19	AEA9839	4	1	0	80
20	AHA8928	5	0	0	100
21	AIA8930	1	4	0	20
22	AJA3359	4	1	0	80
23	AKA5654	5	0	0	100
24	AOA9539	4	1	0	80
25	ARA1577	4	1	0	80
26	ASA1716	5	0	0	100
27	ATA6668	3	2	0	60
28	AUA9975	4	1	0	80
29	AVA2579	5	0	0	100
30	AXA1834	5	0	0	100
	Total	130	20	0	86.66667

Fig. 11 Congestion Charging Main GUI

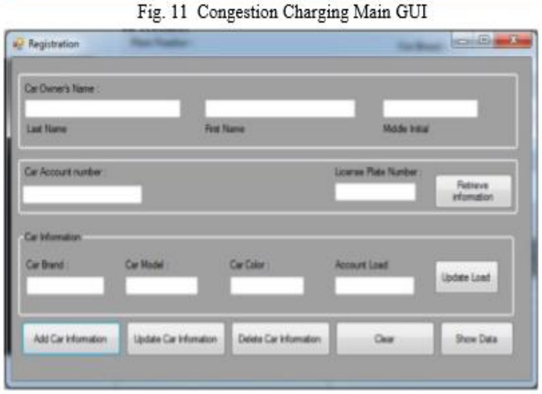

Fig. 12 Database Registration of Car Information

- Car Owner's Name text box – shows the car owner's name.
- Account Number text box – shows the owner's account number.
- Previous Balance text box – shows the previous balance of an account before congestion charge fee deduction.
- Current Balance text box – shows the current balance of an account.
- Entry point text box – shows the entry point where the car entered.
- Exit point text box – shows the exit point where the car exited.
- Car Brand text box – shows the model of the car.
- Car Model text box – shows the model of the car.
- Car Color text box – shows the color of the car.
- Toll Fee text box – shows the congestion charge fee deducted based from the Entry and Exit points.
- Pause button – pauses the video.
- Resume button – resumes the video when paused.
- Start button – starts the system when the necessary Setup is achieved.
- Restart button – restarts the system to load another video or to change the system setup.

Under the Car Information group box, the Setup group box in the rightmost part shows the necessary setup in order to run the GUI. It has the following functions:

- IP Camera button – selects IP camera as video source. • Load Video button – selects recorded video file as video source.
- Entrance Button – selects the identity of the congestion charge as an entry point
- Entry Point groupbox – selects the entry point as A, B or C.
- Exit Button – selects the identity of the congestion charge as an exit point
- Exit Point groupbox – selects the entry point as A, B or C.

Figure 12 shows the car info and registration GUI. It has the following text boxes and buttons and its functions.

- Car Owner's Name

  - Last Name text box – shows/inputs the last name of the owner.

  - First Name text box – shows/inputs the first name of the owner.

  - Middle Initial text box – shows/inputs the middle initial of the owner.
- Car Account Number text box – shows/inputs the account number of the owner
- License Plate Number text box – shows/inputs the plate number of the car.
- Car Brand text box – shows/input the model of the car. • Car Model text box – shows/inputs the model of the car.
- Car Color text box – shows/inputs the color of the car.
- Account Balance text box – shows/inputs the load balance.
- Retrieve Information button – retrieve's the information based on the plate number.
- Add Car Information button – adds a new account to the database.
- Update Car Information button – updates the account of the given plate number. • Delete Car Information button – deletes the account of the given plate number.
- Clear button – clears the text boxes.
- Show Data button – shows the current database.

III. CONCLUSION

This paper demonstrated an automated traffic congestion charging system prototype utilizing OpenALPR with the Tesseract OCR library for Philippine license plates. Although the system needs careful calibration for better accuracy, we have illustrated a proof-of-concept of the feasibility of an ALPR-enabled congestion charging system approach which can achieve an acceptable recognition accuracy for automated congestion charge collection. Future research might include the improvement of technical specifications of video surveillance equipment, i.e. infrared camera for night/dim environment, improving the plate recognition rate, and exploring invariant image frame features which might add to the system's robustness.

REFERENCES

[1] G. Santos, and B. Shaffer, Preliminary results of the London congestion charging scheme. Public Works Management & Policy. 2004 Oct 1;9(2):164-81.

[2] J. Eliasson, L. Hultkrantz, L. Nerhagen, and L.S. Rosqvist. The Stockholm congestion–charging trial 2006: Overview of effects. Transportation Research Part A: Policy and Practice. 2009 Mar 31;43(3):240-50.

[3] S. Du, M. Ibrahim, M. Shehata, and W. Badawy. Automatic license plate recognition (ALPR): A state-of-the-art review. IEEE Transactions on circuits and systems for video technology. 2013 Feb;23(2):311-25.

[4] OpenALPR, https://github.com/openalpr/openalpr, [Online, accessed 10 October 2016]

[5] S. Cheung and C. Kamath, "Robust techniques for background subtraction in urban traffic video." [Online]. Available: https://computation.llnl.gov/casc/sapphire/pubs/UCRL-CONF200706.pdf.

[6] M. Piccardi, "Background subtraction techniques: a review," 2004 IEEE Int. Conf. Syst. Man Cybern. (IEEE Cat. No.04CH37583), vol. 4, pp. 3099–3104, 2004.

[7] I.A, Aljarrah and A.S. Ghorab, "Object Recognition System using Template Matching Based on Signature and Principal Component Analysis," Int. J. Digit. Inf. Wirel. Commun., vol. 2, no. 2, pp. 156–163, 2012.

TABLE VII

Plate Recognition Repeatability Test Accuracy for 20-40 kph for Entrance

Plate Recognition (Exit)					
	Plate No.	True Positive	False Positive	False Negative	Result Percentage
1	AAA6088	3	2	0	60
2	AAB9669	2	3	0	40
3	AAC7171	5	0	0	100
4	AAD1159	5	0	0	100
5	AAJ8762	5	0	0	100
6	AAM1231	5	0	0	100
7	AAN1955	1	4	0	20
8	AAP7860	4	1	0	80
9	AAQ6593	4	1	0	80
10	AAW3081	5	0	0	100
11	AAY3227	5	0	0	100
12	AAZ3545	5	0	0	100
13	ABF7348	4	1	0	80
14	ABG6025	5	0	0	100
15	ABL2841	5	0	0	100
16	ABL8929	3	2	0	60
17	ACA6464	5	0	0	100
18	ADA1716	5	0	0	100
19	AEA9839	4	1	0	80
20	AHA8928	5	0	0	100
21	AIA8930	1	4	0	20
22	AJA3359	4	1	0	80
23	AKA5654	5	0	0	100
24	AOA9539	1	4	0	20
25	ARA1577	4	1	0	80
26	ASA1716	5	0	0	100
27	ATA6668	2	3	0	40
28	AUA9975	4	1	0	80
29	AVA2579	4	1	0	80
30	AXA1834	5	0	0	100
	Total	120	30	0	80

TABLE VIII

PLATE RECOGNITION REPEATABILITY TEST ACCURACY FOR 20-40 KPH FOR ENTRANCE

Plate Detection (Exit)					
	Plate No.	True Positive	False Positive	False Negative	Result Percentage
1	AAA6088	4	1	0	80
2	AAB9669	4	1	0	80
3	AAC7171	5	0	0	100
4	AAD1159	5	0	0	100
5	AAJ8762	5	0	0	100
6	AAM1231	5	0	0	100
7	AAN1955	5	0	0	100
8	AAP7860	4	1	0	80
9	AAQ6593	4	1	0	80
10	AAW3081	5	0	0	100
11	AAY3227	5	0	0	100
12	AAZ3545	5	0	0	100
13	ABF7348	5	0	0	100
14	ABG6025	5	0	0	100
15	ABL2841	5	0	0	100
16	ABL8929	3	2	0	60
17	ACA6464	5	0	0	100
18	ADA1716	5	0	0	100
19	AEA9839	5	0	0	100
20	AHA8928	5	0	0	100
21	AIA8930	5	0	0	100
22	AJA3359	5	0	0	100
23	AKA5654	5	0	0	100
24	AOA9539	4	1	0	80
25	ARA1577	5	0	0	100
26	ASA1716	5	0	0	100
27	ATA6668	4	1	0	80
28	AUA9975	5	0	0	100
29	AVA2579	5	0	0	100
30	AXA1834	5	0	0	100
	Total	142	8	0	94.66667

Design, Development and Implementation of a PID Tuning With Fuzzy Logic Based Flight Controller and Smoothing

Technique of an Unmanned Aerial Vehicle (Quadcopter) Using Arduino Microcontroller

Ralph Laurence G. Visaya[1], Rionel Belen Caldo[2]
[1]Laboratory Custodian, College of Engineering and Computer Studies
[2]Faculty Member, Electronics and Computer Engineering Department
Lyceum of the Philippines University-Laguna (LPU-L), Laguna, Philippines

ABSTRACT

So many researchers proposed control systems for quadcopter in these past few years. In this research paper, the proponent presented the design, development and implementation of a PID Tuning with Fuzzy Logic based flight controller for an unmanned aerial vehicle quadcopter using Arduino microcontroller. The proponents introduce the brief introduction of the quadcopters, its significance and applications. After that, it also introduces the detailed hardware for the quadcopter. The proponent makes use of Arduino IDE for the programming platform of the Arduino and for the simulation of the flight controller. The proponent also makes use of MATLAB Simulink for the PID Tuning; it has been used to test, analyze and compare the performance of the controllers in simulations and the MATLAB Fuzzy Logic toolbox for the trial and error of the pid tuned controller. The Sugeno style of fuzzy inference system was used in the system. Triangular membership functions were used in the control system. Two Fuzzy inputs are constructed namely: the error, and change in error. The proponents also constructed FAM matrix for the rules in the fuzzy logic. The output of the fuzzy logic is normalized in the Dev C++ for fine tuning and getting the accurate and precise outputs. The proponents used the experimentations to validate the functionality and applicability of the designed and developed controller The results of the experiments were satisfactory and it is concluded that it is possible to alleviate quadcopter with PID tuned fuzzy logic based controller.

***Keywords**: unmanned aerial vehicle, quadcopter, Simulink, Mamdani style, Arduino IDE, Fuzzy Logic*

1. INTRODUCTION

This section of research intends to clarify and explain the workings of the research study. It includes the statement of the problem, the objectives and the scope and limitations. This part required to measure the feasibility of the research.

1.1 Background of the Study

In the last few decades, small-scale *unmanned aerial vehicles* have been used for many applications. The need for aircraft with greater maneuverability and hovering ability has led to a rise in quadcopter research. The four-rotor design allows quadcopters to be relatively simple in design yet highly reliable and maneuverable. Research is continuing to increase the abilities of quadcopters by making advances in multi-craft communication, environment exploration, and maneuverability. Quadcopters are a useful tool for university researchers to test and evaluate new ideas in a number of different fields, including flight control theory, navigation, real time systems, and robotics. In recent years many universities have shown quadcopters performing increasingly complex aerial manoeuvers. Swarms of quadcopters can hover in mid-air, fly in formations, and autonomously perform complex flying routines such as flips, darting through hula hoops and organizing themselves to fly through windows as a group. The studies for an unmanned aerial vehicle (UAV) quadcopter modeling and controls have been increased quickly in the recent years. In the quadcopter, it mainly involves complex method such as modeling, control, simulation and parameters tuning etc. Skilled, practiced and expert knowledge is vital in tuning the controller's parameter to get the optimal performances.[3]

1.2 Statement of the Problem

In some studies, the well-known Proportional Integral Derivative (PID) controllers are used in controlling systems like dc motors, quadrotors, etc. Because of its simplicity and efficiency for linear systems, the PID controllers are used for industrial processes. The difference between a measured process variable and a desired set point were calculated by the PID and the manipulated variable is used for adjusting the process of the controller in minimizing error.[1]

However, there are drawbacks in PID controllers. It has been proven that fuzzy logic has more efficient results and precise outcomes. Certainly, fuzzy logic has begun as one of the dynamic areas of study particularly in control applications. It is indeed a very prevailing process when measured models are not accessible and input data are inaccurate. Compared to conventional controlled mechanisms such as PID, fuzzy logic is definitely more useful and accurate. Tuning the parameters of the conventional PID controller is very difficult, poor robustness, so therefore it's difficult to achieve optimal state under field condition in the actual production. For all the problems with the conventional PID controller, fuzzy is the better way to control systems. [1] Fuzzy PID controller method is better method of controlling to the complex and unclear model systems, it can give simple and effective control, play fuzzy control robustness, good dynamic response, rising time, overstrike characteristics. FLC (fuzzy logic control) control has proven effective for complex non linear and imprecisely defined process for which standard model based control techniques are impractical. Fuzzy logic deals with the problems that have vagueness uncertainty and membership function between 0 to1. i.e. if the reliable expert knowledge is not available or if control system is too complex to derive the required decision rules, then some efforts have been made to solve these problems and simplicity the task of tuning parameters and developing rules for the controller. The proponent conducted this study to address the following problems: **a)** *How to implement PID in Fuzzy Logic* and **b)** *how to program PID tuning fuzzy logic in Arduino Microcontroller.*

1.3 Objectives of the Study:

The general objective of this study is to design, develop and implement a pid tuned fuzzy logic based flight controller for a quadcopter using arduino microcontroller. Specifically, the proponent makes a program for the simulation of the PID tuning. The proponent also makes a fuzzy logic algorithim for the fuzzy logic tuning for the PID. The proponent test and experiment the functionality of the sensors needed for the flight controller.

1.4 Significance of the Study:

This study aims to introduce the join forces of the PID and Fuzzy Logic in control systems. The proponent envisions the following to be the main recipient of this research study: a) Students, the professors and the college are to be one to benefit this research study. It could help to enhance their knowledge about arduino, sensors and especially in using MATLAB, PID and Fuzzy Logic. (b) This study may provide innovative knowledge for future developments, enhancements and trends for unmanned aerial vehicle. (c) This study will provide the proponents to have additional source of information and knowledge about the quadcopter, how it works, about control systems, about different algorithms like PID and Fuzzy Logic (d)**.** This study will also be as reference for future researchers who want to pursue this kind of study for their research endeavors.

1.5 Scope and Limitation of the Study

In this study, the proponents will use fuzzy logic for tuning the PID tuned which was simulated in the MATLAB and it will be implemented in Arduino Atmega 2560. The proponents will make use of Sugeno style of fuzzy inference system to fine tune constants for desirable crisp output. In this study, the proponents will be performing experiments and simulations for the accelerometer and gyroscope, barometer sensor, magnetometer sensor, servo motor and camera, testing the functionality of the transmitter and receiver modules with the 2.4GHz remote controller which is being used in RC toys. These experiments and simulations will be integrated to produce the full fuzzy system for the flight control system of the quadcopter. In the fuzzy logic algorithm, the triangular membership functions of the FLC will be constructed with seven (7) classifications such as *"NB", "NM", "NS", "ZO", "PS", "PM"*, "*PB".* The proponents considered only two input variables namely: error (e) and change in error (de). The Fuzzy Associative Memory (FAM) matrix of two-dimensional (2- D) space will be constructed. The three (3) ouputs namely: *Ki, Kp* and *Kd*, served as the crisp output. This will be optimized using trial and error method using DevCpp and Matlab fuzzy logic toolbox. will be fed to the input of the Microcontroller Unit (MPU), which will be used in the flight controller. The code will be uploaded to Arduino IDE (Arduino Atmega 2560) and based on the sensed external temperature condition; the MPU will classify and make decision.

2. METHODOLOGY

2.1 Project Description

This project is a quadcopter that introduces the algorithm of the joined forces of PID and Fuzzy Logic. The frame of this project is made of wood. The center of this project is the microcontroller. The microcontroller is an Arduino Microcontroller. It is very easy and helpful to use. It has also accelerometer and gyroscope. The accelerometer tells the sections of the acceleration vector along the confined coordinate system of the sensor. The gyroscope tells about the angular velocity of the quadcopter. All in all, Accelerometer and Gysroscope sense the motion and operate the motors of the quadcopter. Next is the barometer, it serves as the calculation for the altitude and the temperature. It has also compass sensor which calculate the angle. It also tells the direction of the quadcopter. The quadcopter also has a transmitter and receiver modules. It is installed in order to be controller by a remote controller. The controller used for this is the

2.4GHz Remote Controller that is always used in different RC toys like drones, cars, planes, etc. It also has a camera and servo motor which serves as the pivot joint of the camera so that it will rotate on what is programmed to the servo motor.

2.2 PID Controller

Proportional, integral and derivative Controller (PID), are the most common controllers in the control of industrial processes which has the mathematical development of control theory **Proportional controller (P):** The P proportional part of the PID controller has the task of reducing the rising time. If only a proportional controller is used to control the system, permanent fault occurs in the system. **Proportional-integral controller (PI):** By adding the L Integrator section to the controller P, Permanent fault of the system is deleted. It could also lead to the creation of fluctuations in the system, these fluctuations may be mortal or immortal. So to correct the controller, derivative part is added to it. **Proportional-integral-derivative controller (PID):** Addition of the derivative section enables the controller to predict a kind of system behavior, and before the error rate is increased highly, it corrects the control input. The proponents preferred to use the Ziegler Nichols Tuning method in tuning the PID in the MATLAB.

2.3 Fuzzy Logic Controller

The first step in designing Fuzzy Logic System is to describe fuzzy sets, the information in terms of fuzzy linguistic terms or can be regarded as values equal to 0 or 1 and values within them. The fuzzy membership function is used to assign values with respect to fuzzy sets and its degrees of membership. Moreover, the most commonly used technique requiring four parameters to be specified with respect to its x and y coordinates is called Trapezoidal Membership Function. Also, the Triangular Membership Function can be used in the system which requires three parameter specifications. The proponents preferred to use triangular membership function in this proposed [14].

2.4 PID Fuzzy Logic Structure

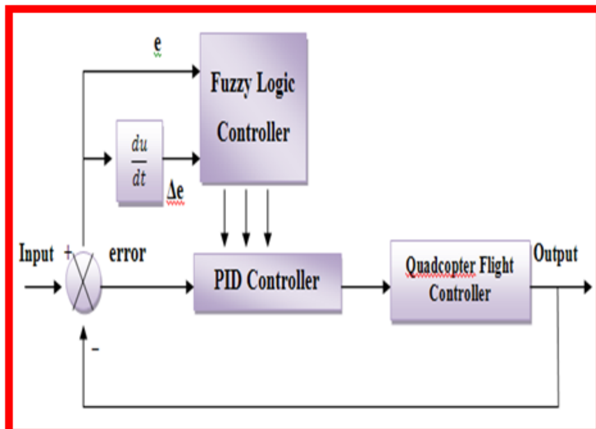

Figure 1: *PID Fuzzy Logic Structure Block Diagram*

In the figure 1, it shows the block diagram of the PID fuzzy logic structure. The fuzzy logic controller takes the system error denoted by **"e"** and the change in error denoted by **"Δe".** The output of the fuzzy logic will goes to the PID controller namely: Kp, Ki and Kd. The output of the PID will goes to the flight controller of the quadcopter.

2.5 Arduino IDE

This project is a software controlled microcontroller based system, which makes it intelligent, independent of operator supervision and can adapt to a restricted set of changed requirements without any hardware change. The software for this project was developed using a simple high level language in C++. It also used ARDUINO IDE sketchbook for PC to system interface. The software controls the input display of the device and manipulates its polarity depending on the parameters of the equipment engaged through it. The Arduino IDE is an open-source Arduino Software it provides an easy way to write a code and upload it to the board, it runs on Windows, Mac OS X, and Linux. The environment is written based on Processing, Java, and other open-source software's. The software can be used with any Arduino board.

2.6 Hardware Block Digram

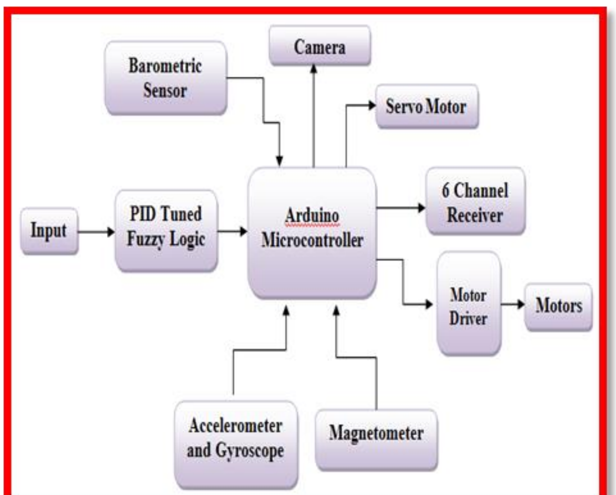

Figure 2: *Hardware Block Diagram*

In the figure above, it shows the block diagram of the full system of the quadcopter. The receiver signal is being transmitted to the RC controller in order to control the quadcopter then the input will be decided on the PID Fuzzy Logic controller after that after the output of the controller are to be fed to the Arduino Microcontroller. The Barometric Sensor can be able to sense and tell about the altitude of the quadcopter, the accelerometer can be able to tell about the angle, the acceleration and the speed, the x,y and z channel of the quadcopter. The Magnetometer tells the direction and the magnitude of the quadcopter. The servo motor helps to roate the camera in 180 degrees. Then the motor reacts depending on the input.

3. DESIGN CONSIDERATIONS

3.1 The Prototype

Figure 3: The Quadcopter

In figure 3, it shows the actual prototype of the system. It is the quadcopter arranged with a PID tuned fuzzy logic controller. The flight controller of the quadcopter is an Arduino Microcontroller based. The connection in the controller comprises of the Triple Axis Acclerometer and Gysroscope, Magnetometer, Barometric Pressure Sensor, Servo motor for controlling the rotation of the camera, it has also 6-ch transciever just for recognizing by the 2.4GHz RC Toy Remote Controller. The frame of the quadcopter is made of wood. The motor of this prototype is a Brushed DC motor.

3.2 System Sensors:

3.2.1 MPU6050 (Triple Axis Accelerometer and Gyroscope)

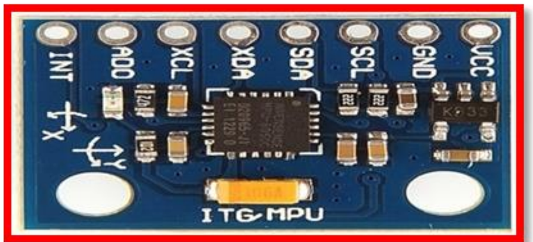

Figure 4: *MPU6050*

In this research, the proponents make use of the the MPU6050 Triple Axis Acclerometer and Gyroscope in the system. The following reasons why use this sensor:

a. Averaging the data that comes from accelerometers and gyroscopes can produce a better estimate of orientation than obtained using accelerometer data alone.
b. Also has Digital Motion Processor (DMP)
c. Economical and cheaper than the separate modules for the accelerometer and gyroscope.
d. It is compatible to Arduino Microcontroller.

3.2.2 BMP180 (Barometric Pressure Sensor)

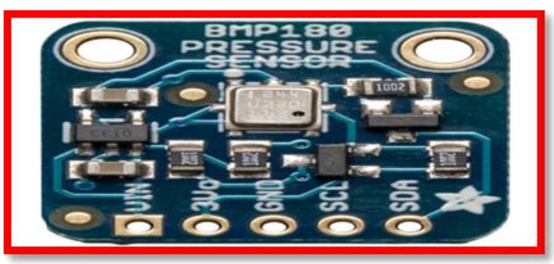

Figure 5: BMP180

This precision sensor is the best low-cost sensing solution for measuring barometric pressure and temperature. Because pressure changes with altitude you can also use it as an altimeter! The sensor is soldered onto a PCB with a 3.3V regulator, I2C level shifter and pull-up resistors on the I2C pins. It is also

3.2.3 HCM5883L (Magnetometer)

Figure 6: *HCM5883L*

HMC5883L is a triple axis magnetometer. Magnetometers have a wide range of uses. The axes are labeled as X, Y, and Z. An output is provided for each of these axes that describe the position of these axes relative to the earth's magnetic field.

3.3 Fuzzy Logic Simulation:

Figure 7: *FIS EDITOR*

In the figure, it shows the Fuzz Logic Toolbox. It comprises of the 2 inputs and 3 outputs for the tuning the PID. In the center box, it appears the name of the file and the type of fuzzy logic interference system used in the system. In the system, the proponents used the Sugeno Style FIS System.

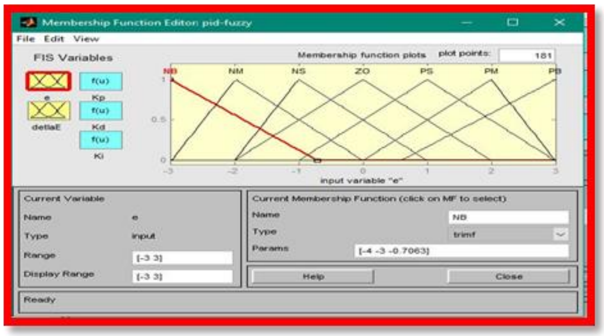

Figure 8: *Error (Input 1)*

In the above figure shows the Input error of the fuzzy logic. It also shows the Membership functions. In the FIS, the proponents make use of the triangular membership function and comprises of 7 parameters namely: "NB, NM, NS, ZO, PB, PM, PS".

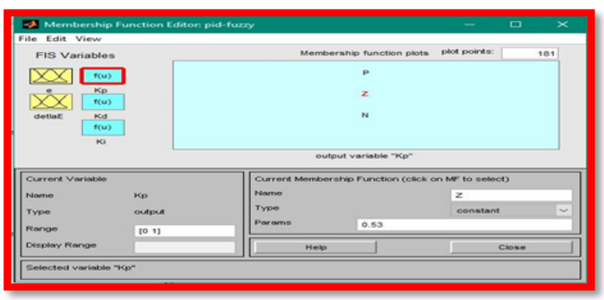

Figure 9: *The Output (Kp)*

The figure shows the Kp output. It also shows the output of the sugeno style fuzzy interference system and the membership functions and its parameters namely: P, Z and N. This is only for the Kp and the others (Ki and Kd) are in the below part of the Kp.

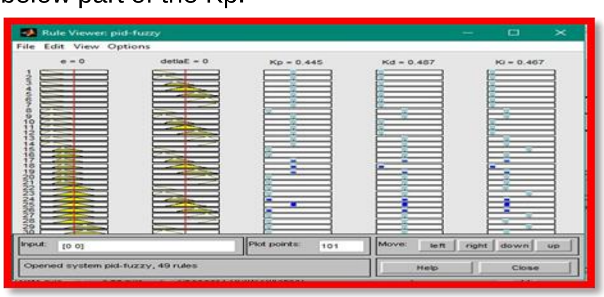

Figure 10: *Rule Editor*

This is the part where rules can be add and deleted. This is also where rules are constructed. The rules comprise about 49 rules.

Figure 11: *Rule Viewer*

This figure shows the rule viewer. The values of the inputs can be changed and see what values would be the output.

4. EXPERIMENTS AND ANALYSIS OF RESULTS

The point of this chapter is to show the analysis and interpretation of data for series of experiments. The proponents simulated the tuning of PID in MATLAB through the use of Simulink. Then after the tuning, the output of the simulation had been the input for another tuning in the fuzzy logic toolbox also in MATLAB.

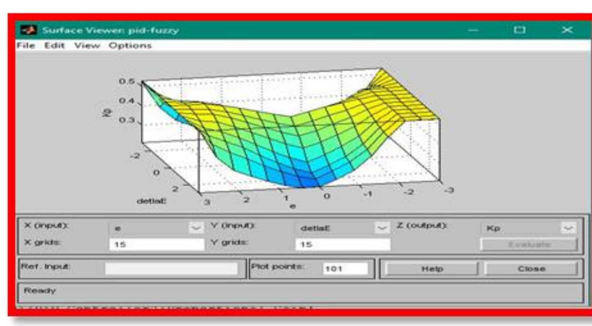

Figure 12: *Surface View **Kp***

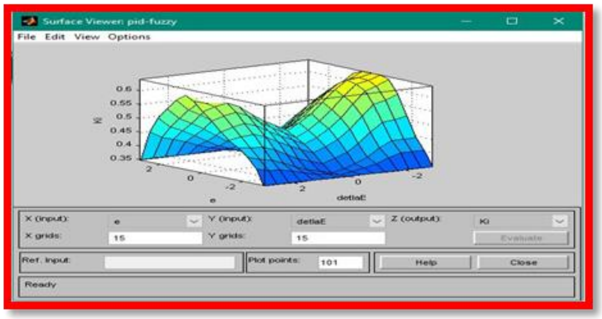

Figure 13: *Surface View **Kp***

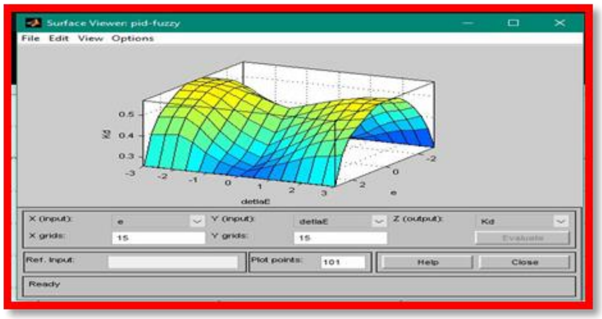

Figure 14: *Surface View **Kd***

In figures 12 – 14, it shows the surface view of the PID Gains, Ki, Kd and Kp. The colors changes according to the output values. In the picture, the highest value is color yellow and the lowest value is color blue. In the evelation map, it will help to see what is the highest and lowest.

4.1 Statistical Tool

4.1.1 Smoothing Technique

When data collected over time displays random variation, smoothing techniques can be used to reduce or cancel the effect of these variations. When properly applied, these techniques smooth out the random variation in the time series data to reveal underlying trends. Inherent in the collection of data taken over time is some form of random variation. There exist methods for reducing of canceling the effect due to random variation. An often-used technique in industry is "smoothing". This technique, when properly applied, reveals more clearly the underlying trend, seasonal and cyclic components.

There are two distinct groups of smoothing methods

a. Averaging Methods
b. Exponential Smoothing Methods

The proponents used the Averaging Method in observing and proving the the PID Fuzzy Logic is much more accurate than PID.

Table 1: *Settling Time Measured in Altitude Control*

Estimated = 1 secs			
	Trials	PID	Fuzzy PID
Altitude Control	1	1.412	0.983
	2	1.621	0.894
	3	1.437	0.872
	4	1.174	0.972
	5	1.452	0.962
	6	1.52	0.967
	7	1.749	0.876
	8	1.64	0.982
	9	1.452	0.986
	10	1.426	0.975

In the table, it shows the comparison of the settling time between those two controllers. The proponents tested the time by using a stopwatch and mesasured the time response of the quadcopter in the part of altitude control. The altitude control is the time the quadcopter responded in lifting or flying upward direction.

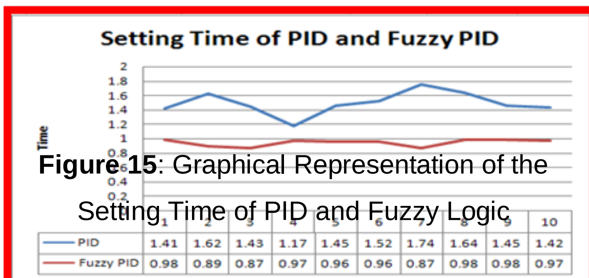

Figure 15: Graphical Representation of the Setting Time of PID and Fuzzy Logic

In this figure, show the graphical comparison of the two controllers. Fuzzy PID has the faster time response than PID alone. The proponents estimated 1 sec time response for the controller and the fuzzy pid the only one matches the estimated time response in the quadcopter.

Table 2: *Percentage Error*

PID Error in %	Fuzzy PID Error in %
41.2	-1.7
62.1	-10.6
43.7	-12.8
17.4	-2.8
45.2	-3.8
52	-3.3
74.9	-12.4
64	-1.8
45.2	-1.4
42.6	-2.5

The table shows that the PID controller has the highest percentage error while the Fuzzy Logic PID has the lowest percentage error. Thus, it is being resulted that the fuzzy logic is reliable.

4.2 Statistical Analysis Using Moving Average Smoothing Technique

Table 3: *Using Smoothing Technique / Moving Average*

Moving Average	PID	Fuzzy PID
1	1.4192	0.9366
2	1.4408	0.9334
3	1.4664	0.9298
4	1.507	0.9518
5	1.5626	0.9546
6	1.5574	0.9572

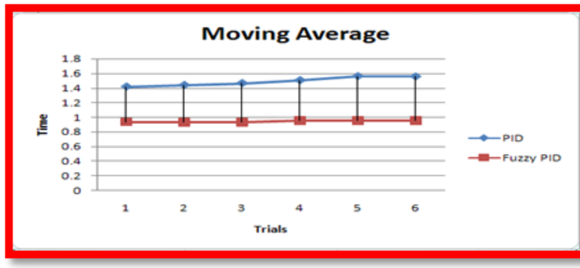

Figure 16: *Graphical Representation for the Statistcal Analysis for PID and Fuzzy Logic*.

In the figure and the table show the analysis for using the moving average smoothing technique. The proponents used this to find and to see that the data is more accurate than taking down the time response in 10 trials. It shows that the PID also is slower than Fuzzy PID.

5.CONCLUSION

After the series of experimentation for the hardware parts and the software parts for the flight controller of the quadcopter. the results were carefully studied and interpreted accordingly. Based on the data analysis and reports, the output of the controller was favorable and satisfactory because that fuzzy pid can be implemented on arduino microcontroller and at the same time the time response for different controls the atltitude, yaw, throttle, the experimentation and results, it is evident and proof that the time measured in PID and Fuzzy PID, which prove that the research is reliable and accurate because of the difference in their time responses. The PID is slower than of the Fuzzy PID. It also convenient to use because of its minimal delay time of the controller.. Fuzzy PID Based Controller responds in a very least possible time. Thus, the time response is strong and almost in the , the system is reliable. The research is properly and accurately working which satisfied the objectives of this study.pitch of the quadcopter meets the expected time response ranging from 0.89 to 0.98 secs with the maximum requirement of 1 secs. Analyzing the tables and graphs in using the Smoothing Technique, it can see clearer that the Fuzzy PID is faster than the PID so it must be used in the quadcopter specially when the quadcopter used for racing, for surveillance, etc.

6. ACKNOWLEDGEMENT

The proponent acknowledge Jesus Christ, who always give them knowledge, wisdom and strength in this research. He would like to thank Engr. Rionel B. Caldo for his continuous support and advices as the preferred thesis adviser.

7. REFERENCES

[1] Ralph Laurence G. VIsaya, et al, DESIGN AND DEVELOPMENT OF FUZZY LOGIC ALGORITHM WITH VARYING SETPOINTS FOR FAN SPEED CONTROL SYSTEM USING ARDUINO MICROCONTROLLER, 2014

[2] Mustafa R Abuzeid, Ramadan AElmoude, Nasr E Shtawa. A novel Strategy for Synchronizing of Two DC Motors, Al-azhar Engineering Ninth International Conference, April 12-14, 2007.

[3] https://en.wikipedia.org/wiki/Quadcopter

[4] Jitender Kaushal, Comparative Performance Study of ACO & ABC Optimization based PID Controller Tuning for Speed Control of DC Motor Drives, Issue JUNE, 2012.

[5] Bin Adzmi K .A, Motor Speed Controller Using Fuzzy Logic Method for PCB Drilling Operation, Issue 4 November, 2008.

[6] S.Z. He. S. Tan, F.L.Xu, ''Fuzzy self-tuning of PID controllers"", fuzzy set and Systems, 56(1993), 37-46.